Why do I need four search engines?

Martin Holmes and Joey Takeda
University of Victoria Endings Project

University of Victoria

Project Endings

- How to complete, publish and walk away from your digital edition project…

University of Victoria

Project Endings

- How to complete, publish and walk away from your digital edition project…
- ...and have it last for 50 years.

The Projects

- The Map of Early Modern London ↗
- Le Mariage sous l'ancien régime ↗
- The Nxaʔamxcín Dictionary Database
- **The Robert Graves Diary** ↗
- Mapping Keats's Progress ↗
- The Scandinavian-Canadian Studies Journal ↗

University of Victoria

Diary of Robert Graves 1935-39 and ancillary material

Search Graves Diary Collection

Search for: (Enter keywords separated by spaces)

Search

Special characters

Match: ALL Keywords ANY Keyword

Returns/Page 10

Order By Date ascending

Include:

- Abstracts
- Diary Entries
- Enclosures
- Log Entries

Date Range:

	Day:	Month:	Year:
Begin Search:	22	February	1935
End Search:	6	May	1939

Browse Diary Entries

Day: 22

Month: February

Year: 1935

View

Browse Abstracts

Month: February

Year: 1935

View

Original site: 2003
Endings rebuild: 2017

Rebuild is XHTML5 but aims to replicate the original design.

Previous work

Arneil, Stewart and Martin Holmes. 2017. “Archiving form and function: preserving a 2003 digital project.” DPASSH Conference 2017: Digital Preservation for Social Sciences and Humanities, Brighton, UK, 14th June 2018.

Holmes, Martin. 2017. “Selecting Technologies for Long-Term Survival.” SHARP Conference 2017: Technologies of the Book, Victoria, BC, Canada, 10th June 2017. [PDF].

Holmes, Martin and Joseph Takeda. 2017. “Beyond Validation: Using Programmed Diagnostics to Learn About, Monitor, and Successfully Complete Your DH Project.” Digital Humanities 2017 Conference, Montreal, Canada, 1th August 2017. [PDF]

Endings Principles

- Endings principles cover five components of a digital project:
  - DATA
  - PRODUCTS
  - PROCESSING
  - DOCUMENTATION
  - RELEASE MANAGEMENT

Endings Principles

- Endings principles cover five components of a digital project:
  - DATA
  - **PRODUCTS**
  - PROCESSING
  - DOCUMENTATION
  - RELEASE MANAGEMENT

Principles for Products

Products are the project outputs intended for end-users, typically in the form of websites or print documents. The following principles apply to products intended for the web:

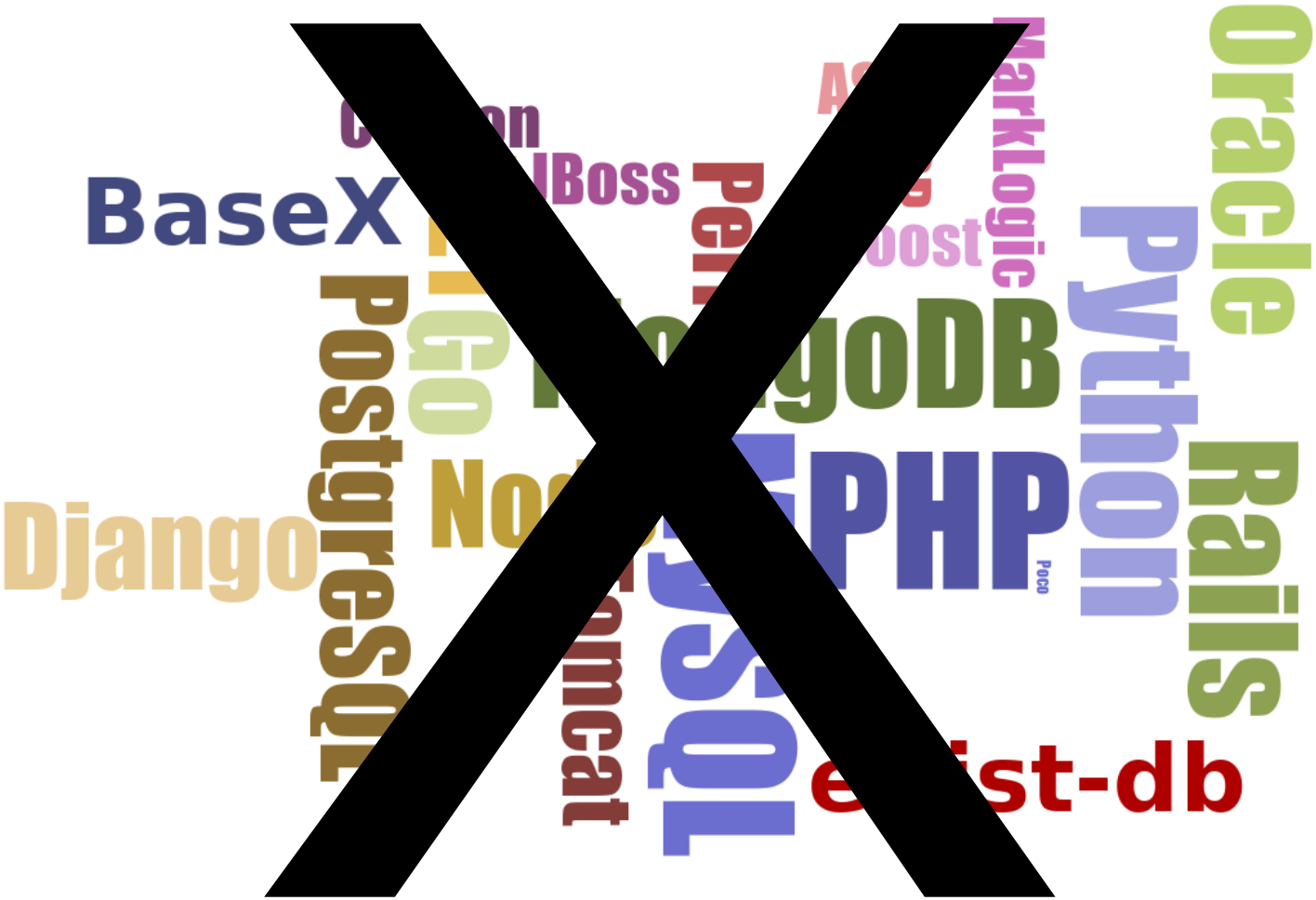
Principles for Products

Products are the project outputs intended for end-users, typically in the form of websites or print documents. The following principles apply to products intended for the web:

2.1 No dependence on server-side software: build a static website with no databases, no PHP, no Python.


Oracle
MarkLogic
ASP
JSP
Cocoon
JBoss
Perl
Boost
Python
BaseX
NETGO
MongoDB
Rails
PostgreSQL
eXist
PHP
Node
Poco
Django
MySQL
CouchDB
Tomcat
eXist-db

It’s ALL GOING AWAY.



Principles for Products

2.2 No boutique or fashionable technologies: use only standards with support across all platforms, whose long-term viability is assured. Our choices are *HTML5*, *JavaScript* and *CSS*.

Principles for Products

2.3 No dependence on external libraries:
no JQuery, no AngularJS, no Bootstrap.

Principles for Products

2.4 No query strings: every entity in the site has a unique page with a simple URL.

Principles for Products

2.6 Massive redundancy: every page contains all the components it needs, so that it will function without the rest of the site if necessary, even though this means duplicating information across the site.

Principles for Products

2.7 Relentless validation: every site build involves validation of all input data (XML) and all output code (HTML5, JavaScript, CSS).

Principles for Products

2.8 Inclusion of data: every site should include a documented copy of the source data, so that users of the site can repurpose the work easily.

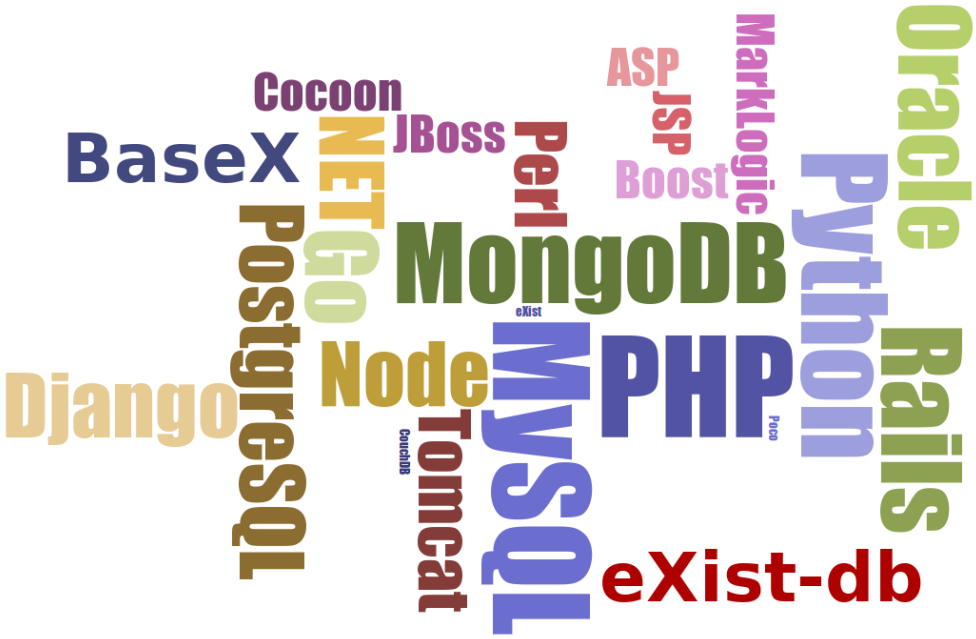
Endings principles document:
https://raw.githubusercontent.com/projectEndings/Endings/master/principles.txt

Result:

- A completely static site consisting of HTML (XHTML5), JavaScript and CSS.
- Graceful degradation for JS and CSS.
- Each page is *coherent*, *consistent* and *complete*.
- The site works on any webserver, or from a local drive, USB stick, DVD, etc.

What about search?

- All sites need a half-decent search engine.
- But search engines typically depend on one or more of:



Pact with the devil clause:

- 2.9 Once a fully-working static site is achieved, it may be enhanced by the use of other services such as a server-side indexing tool (Solr, eXist) to support searching and similar functionality.

Search engine #1

Search engine #1

- Searches the HTML, not the XML. ✔
- Provides faceting based on metadata in the HTML. ✔
- Provides keyword-in-context results. ✔

Diary of Robert Graves 1935-39 and ancillary material

Search Graves Diary Collection

Search for: (Enter keywords separated by spaces)

Search

Special characters

Match: ALL Keywords ANY Keyword

Returns/Page 10

Order By Date ascending

Include:

- Abstracts
- Diary Entries
- Enclosures
- Log Entries

Date Range:

	Day:	Month:	Year:
Begin Search:	22	February	1935
End Search:	6	May	1939

Browse Diary Entries

Day: 22

Month: February

Year: 1935

View

Browse Abstracts

Month: February

Year: 1935

View

But...

- exist-db is a small open-source project. ✘
- It needs Java. ✘
- Sysadmins don’t know or care about it. ✘
- It’s likely to require periodic updates, rebuilds, migration and other TLC. ✘

But...

- exist-db is a small open-source project. ✘
- It needs Java. ✘
- Sysadmins don’t know or care about it. ✘
- It’s likely to require periodic updates, rebuilds, migration and other TLC. ✘
- NOT GOOD ENOUGH!

Search engine #2

Google

Search engine #2

Google™

Search engine #2

Google™

- Easy to set up ✔
- Familiar user interface ✔
- Cheap on resources (for us) ✔
- Provides keyword-in-context results ✔

Diary of Robert Graves 1935-39 and ancillary material

Google Search

bullfight

Web | Image

About 61 results (0.35 seconds)

Sort by: Relevance

Enclosure – **Bullfight** Ticket
https://graves.uvic.ca/diary_1936-06-07_01_enc.html

Enclosure – **Bullfight** Ticket. Annotated markup · Full-sized Image · Gallery Scan. (p. 1 of 1). Scan for 1936-06-24. Enclosure – San Juan Fiesta programme.
Structured data

Diary of Robert Graves 1935-39 and ancillary material: Entry for ...
https://graves.uvic.ca/diary_1935-07-07.html

Belmonte, Lalanda, Cayetano (Niño de la Palma)2. Fair haired girl slightly hurt by taxi in Calle Arabi3; on the way to Alhambra. Juan pessimistic about **bull-fight**.
Structured data

But...

- Trust Google?

But...

- Trust Google?
- NO WAY!

But...

- Trust Google?
- NO WAY!
- Works today, breaks tomorrow. ✘

But...

- Trust Google?
- NO WAY!
- Works today, breaks tomorrow. ✘
- Exists today, disappears tomorrow. ✘

But...

- Trust Google?
- NO WAY!
- Works today, breaks tomorrow. ✘
- Exists today, disappears tomorrow. ✘
- Free today, costs tomorrow. ✘

But...

- Trust Google?
- NO WAY!
- Works today, breaks tomorrow. ✘
- Exists today, disappears tomorrow. ✘
- Free today, costs tomorrow. ✘
- Depends on a stable domain. ✘

But...

- Trust Google?
- NO WAY!
- Works today, breaks tomorrow. ✘
- Exists today, disappears tomorrow. ✘
- Free today, costs tomorrow. ✘
- Depends on a stable domain. ✘
- NOT GOOD ENOUGH!

So who do we trust?

Search engine #3

- University Library’s Solr instance.

How it works

- Graves site build process:
  - …
  - TEI XML → XHTML5
  - …
  - XHTML5 → Solr index files (XML)
- Solr index files → Solr administrator
- Application queries Solr instead of eXist indexes.

But...

- Solr only accepts connections from a known IP address. ✘
- Solr itself may go away, or change beyond recognition. ✘
- *The library itself* may go away, or change beyond recognition. ✘
- NOT GOOD ENOUGH!

Search engine #4

Search engine #4

- Standalone all-JavaScript no-backend keyword search with stemming and relevance scoring

How it works (1)

- Graves site build process:
  - …
  - TEI XML → XHTML5
  - Tokenize HTML body text.
  - Stem the tokens (Porter stemming).
  - For each token, create a JSON file named for the token.
  - In the JSON file, place a pointer to each document containing the token, with a score for the number of times it occurs in the document.
  - = 11,776 files, 23.2 MB

```
{ "token" : "childish",
  "instances" :
  [
    { "docId" : "diary_1938-03-18",
      "docTitle" : "Entry for 1938-03-18",
      "docType" : "diaryentry",
      "docStartDate" : "1938-03-18",
      "docEndDate" : "1938-03-18",
      "count" : 1 },

    { "docId" : "abstract_1938-03",
      "docTitle" : "Abstract for March 1938",
      "docType" : "abstract",
      "docStartDate" : "1938-03-01",
      "docEndDate" : "1938-03-31",
      "count" : 1 }
  ]
}
```

How it works (2)

- Search page:
  - User types in keywords.
  - Keywords are stemmed by JavaScript.
  - For each unique token, retrieve the JSON file named for it.
  - Combine the scores for each document across the tokens.
  - Additional filtering by date and document type.
  - Present results ordered by score.

Search Graves Diary Collection

Search for: (Enter keywords separated by spaces)
For proper names, use initial capitals.

love

Search

Include:

- [x] Abstracts
- [x] Diary Entries
- [x] Enclosures
- [x] Log Entries

Date Range:

	Day:	Month:	Year:
Begin Search:	1	January	1937
End Search:	6	May	1939

Searched for: love
Documents found: 48

- Enclosure – Letter to RG and LR from Karl Goldschmidt 1938-10-17 (Score: 6)
- Enclosure – 5-page letter to RG from Jenny in Liverpool 1938-12-12 (Score: 5)
- Enclosure – Letter to LR from Margaret Russell 1938-09-03 (Score: 4)
- Enclosure – Letter to RG from Ros Graves 1939-01-12 (Score: 3)
- Enclosure – Postcard to RG from David Graves 1938-03-31 (Score: 3)
- Enclosure – Letter from Catherine Nicholson, signed Kate 1937-11-30 (Score: 2)
- Enclosure – Letter from David Graves 1938-10-01 (Score: 2)
- Enclosure – Letter to RG from Jenny Nicholson 1937-08-05 (Score: 2)
- Entry for 1939-03-07 (Score: 2)
- Enclosure – Letter to RG and LR from Sam Graves 1938-11-04 (Score: 2)
- Entry for 1938-04-08 (Score: 2)

Pros and cons

- Lightning fast ✔
- Works anywhere ✔
- Usable and effective ✔
- No keywords-in-context ✘
- Only practical for small projects ✘

Graceful degradation

If existdb then existdb ...

Else if Solr then Solr ...

Else if Google then Google ...

Else JavaScript

Insane?

- **exist-db** we're doing anyway.
- **Solr** invokes institutional support / responsibility / attention.
- **Google** is easy – why not?
- **JavaScript** is easy (once coded) and bulletproof.